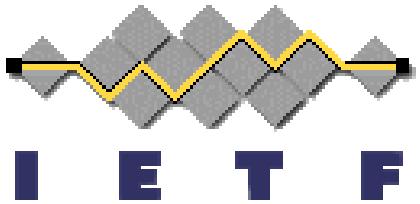
NFS Version 4 Security Update

Mike Eisler
Sun Microsystems, Inc.
mre@Eng.Sun.Com

45th IETF
Oslo
July 11-16, 1999



NFS V4 Working Group

Contents

NFS V2/V3 security draft update

Summary of issues raised in San Jose

A lightweight security mechanism for NFS

Other security issues

Straw polls

NFS V2/V3 Security Draft Update

- The specification in draft-ietf-nfsv4-nfssec-01.txt, “NFS Version 2 and Version 3 Security Issues and the NFS Protocol’s Use of RPCSEC_GSS and Kerberos V5” was originally offered as an Informational RFC to IETF
- IESG reviewed it and decided it should be a standards track document reviewed by the NFS V4 WG
- After WG review, IESG referred the document to IANA for review
  - IANA asked for an “IANA Considerations” section to discuss registration of the pseudo flavors for NFS V3’s security negotiation in version 3 of the MOUNT protocol
- Once these changes were made, the WG blessed them, and IESG approved the document for publication
- In June, 1999, the RFC editor published draft-ietf-nfsv4-nfssec-01.txt is as RFC 2623
- While RFC 2623 is a product of the NFS V4 WG, it is not a mandate for NFS V4’s security model.

Summary of security issues raised in San Jose WG in March, 1999

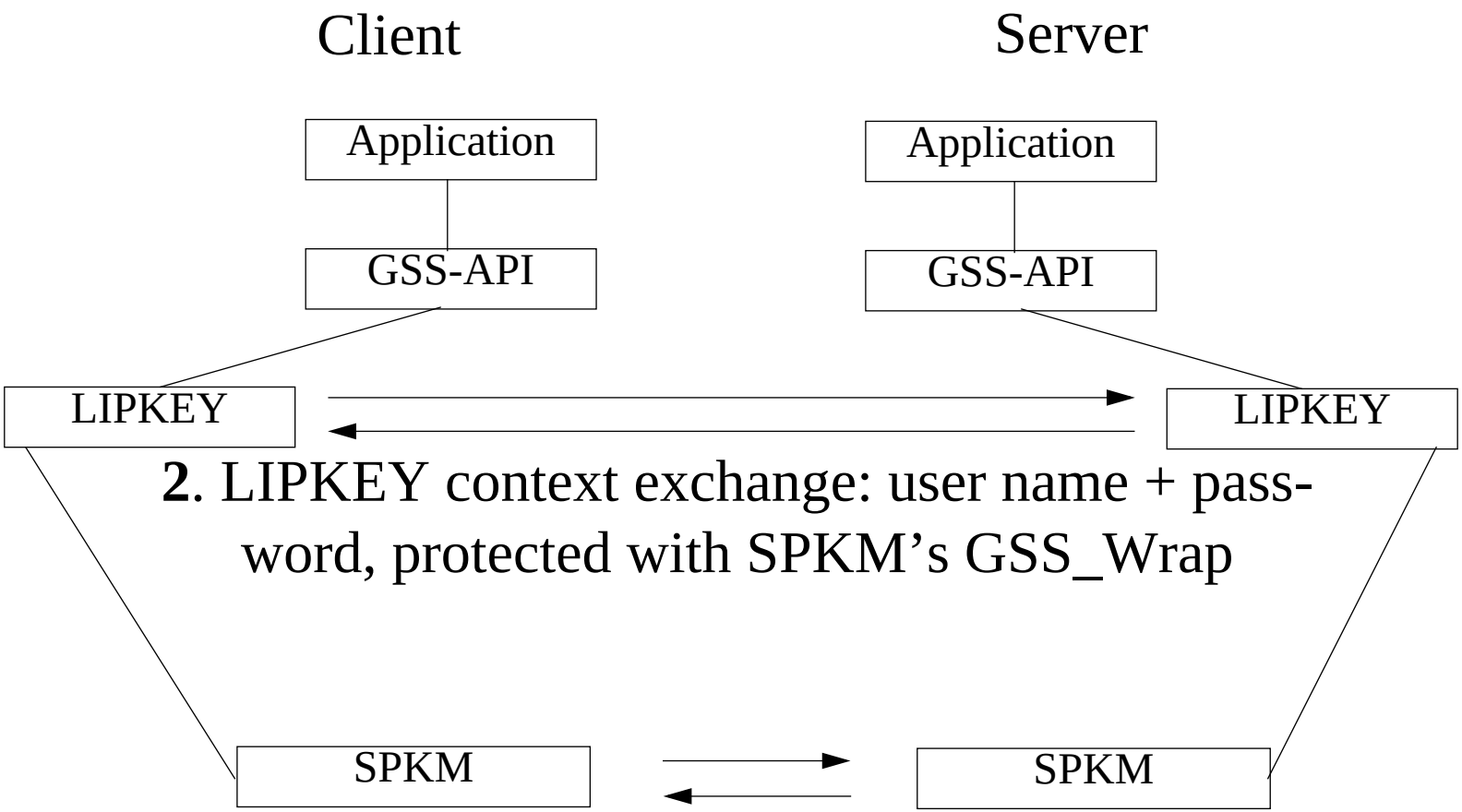
Reference: http://playground.sun.com/pub/nfsv4/nfsv4-wg-archive/1142.html

- Consensus reached on
  - Using ONC RPC as the transport for NFS V4
  - Using RPCSEC_GSS as the security framework for NFS V4
- Controversy on using Kerberos V5 as a mandatory to implement (though not “mandatory to use”) security mechanism under RPCSEC_GSS
- Interest in the using draft-ietf-cat-lipkey-XX.txt as one of the mandatory to implement mechanisms
- Suggestions for using TLS and IPSEC
  - IPSEC doesn’t support multiple users per TCP connection
  - No one has volunteered to design an NFS over TLS framework

LIPKEY: A Low Infrastructure Public KEY security mechanism

- LIPKEY has been updated: draft-ietf-cat-lipkey-01.txt
- LIPKEY shares the typical SSL/TLS model
  - Using SSL/TLS, a web browser takes server's public key (from its certificate), and encrypts a session key with it.
  - Session key sent to server
  - Client and server now have a secure channel without needing a user certificate
  - Client then sends a user name and password, encrypted with the session key
  - Server authenticates client

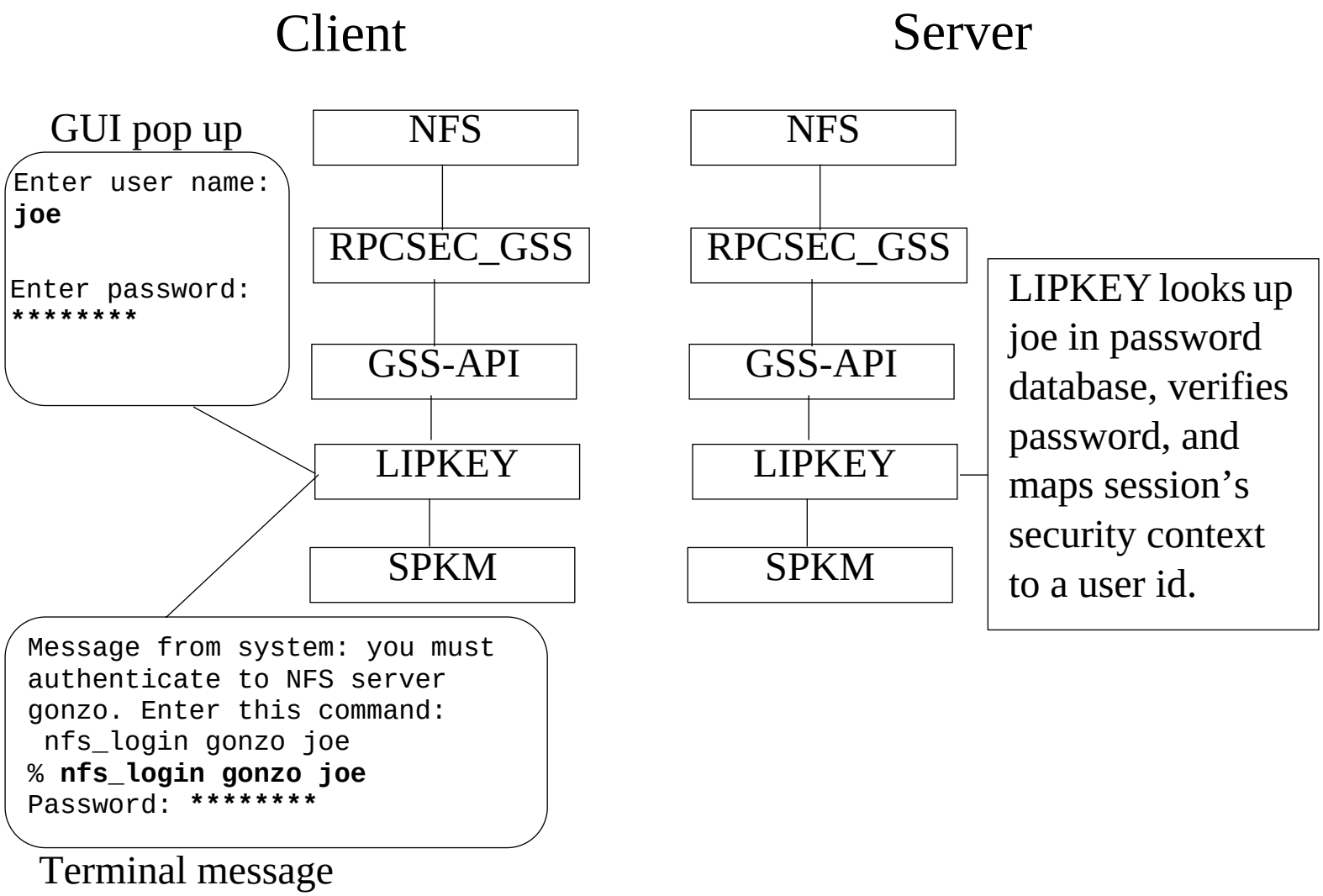
Brief overview of LIPKEY



2. LIPKEY context exchange: user name + password, protected with SPKM's GSS_Wrap

1. SPKM-1 unilateral (server only) authentication context exchange. Initiator is anonymous with no certificate required.

NFS V4 and LIPKEY



Other security issues

uid and gid representation

- “mapping 32 bit uids to strings is costly”
- “so is mapping 32 bits to 128 bit UUIDs”
- apparently no consensus that the NFS V2/V3 model needs fixing

proxies

- security discussion can’t proceed until we agree on the proxy model
- model 1: client is unaware that the NFS server is a proxy
    - trivially implemented by the NFS server exporting its NFS mounted file systems
    - client authenticates only to proxy.
    - This is not end to end authentication
- model 2: client is aware that the NFS server is a proxy
    - requires changing NFS V4 protocol to let client indicate that the request is to use the server as a proxy
    - end to end authentication possible

Straw Polls

Should NFS V4 use TLS for security?

- In addition to RPCSEC_GSS?
- Instead of RPCSEC_GSS?
- Is anyone willing to sign up to design it?

Should NFS V4 specify Kerberos V5 as mandatory to implement?

Should NFS V4 specify LIPKEY as mandatory to implement?

Is the NFS V2/V3 32-bit uid/gid model broken in that it

- is uncontrolled?
- allows too few unique identifiers?

If so, should we fix it NFS V4?

If so, should we use

- strings of form user@DNS_domain?
- UUIDs?
- something else?
- two or more of the above?
- all of the above?